Stereo Event Lifetime and Disparity Estimation for Dynamic Vision Sensors

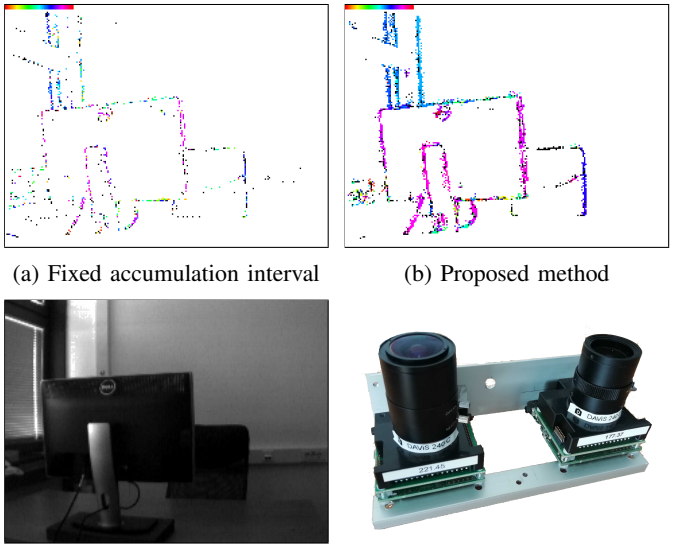
Antea Hadviger, Ivan Marković, Ivan Petrović*

***Abstract*—Event-based cameras are biologically inspired sensors that output asynchronous pixel-wise brightness changes in the scene called events. They have a high dynamic range and temporal resolution of a microsecond, opposed to standard cameras that output frames at fixed frame rates and suffer from motion blur. Forming stereo pairs of such cameras can open novel application possibilities, since for each event depth can be readily estimated; however, to fully exploit asynchronous nature of the sensor and avoid fixed time interval event accumulation, stereo event lifetime estimation should be employed. In this paper, we propose a novel method for event lifetime estimation of stereo event-cameras, allowing generation of sharp gradient images of events that serve as input to disparity estimation methods. Since a single brightness change triggers events in both event-camera sensors, we propose a method for single shot event lifetime and disparity estimation, with association via stereo matching. The proposed method is approximately twice as fast and more accurate than if lifetimes were estimated separately for each sensor and then stereo matched. Results are validated on real-world data through multiple stereo event-camera experiments.**

***Index Terms*—event-based cameras, stereo vision, event lifetime estimation, disparity estimation**

I. Introduction

Event-based vision sensors [1], such as the dynamic vision sensor (DVS) [2] and dynamic and active-pixel vision sensor (DAVIS) [3], are relatively novel biologically inspired cameras that output pixel-wise changes in brightness intensity in the scene, which are referred to as *events* due to their asynchronous nature. Since event-based vision sensors do not record the brightness intensity of an entire scene at a fixed frame rate, but offer a sparse data stream reporting only changes in the intensity, they have significantly lower power consumption and bandwidth requirements than standard cameras still widely used in computer vision. Furthermore, event-based cameras operate with low latency and with a temporal resolution of a microsecond, thus avoiding the problem of motion blur that makes standard cameras unusable in highly dynamic scenarios. Also, event-based cameras have the advantage of a high dynamic range (HDR) of 140 dB, compared to the 60 dB of standard cameras, making them an interesting sensor for tasks with challenging scene illumination and environmental conditions, often encountered by autonomous mobile robots. A recent survey on event-based vision applications and methods can be found in [4] and an example of a processed DVS output can be seen in Fig. 1.



(a) Fixed accumulation interval (b) Proposed method

(c) Grayscale image (d) Experimental stereo setup

Fig. 1: Stereo DVS events – using a fixed event accumulation time of 10 ms yields a sparse event disparity map prone to noise, while disparity obtained with the proposed method retains scene structure and successfully filters noise. Color bar in the top left corner indicates disparity, i.e., depth. Black pixels indicate no available disparity.

Standard computer vision algorithms that operate on image intensity frames cannot be directly applied to the fundamentally different event stream of event-based vision sensors. Recently, significant efforts have been made in the area of intensity image reconstruction from events [5], [6]; however, for some applications complex image intensity reconstruction is not needed. If the incoming events are simply accumulated over time, they can be represented as a frame and used as an input to slightly altered standard image processing algorithms. Accumulation can be done directly by choosing a fixed time interval and this approach has been used for tracking and optical flow estimation [7], [8]. Events can also be accumulated by choosing a certain number of events to be shown in each generated frame, as shown in [9] for simultaneous localization and mapping (SLAM). Since the frequency of incoming events depends on the amount of motion of the

This work has been supported by the FLAG-ERA JTC 2016 and the Ministry of Science and Education of the Republic of Croatia under the project *Rethinking Robotics for the Robot Companion of the Future* (RoboCom++) and the Croatian Science Foundation under contract No. I-3137-2019. The research has been carried out within the activities of the *Centre of Research Excellence for Data Science and Cooperative Systems* supported by the Ministry of Science and Education of the Republic of Croatia.

*Authors are with the University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics, Croatia. {antea.hadviger, ivan.markovic, ivan.petrovic}@fer.hr

scene and the camera itself, an accumulation interval that is too long will produce blurred images, while a short interval will cause loss of structure in the scene, thus not allowing correct interpretation. Given that, regardless of the strategy, event accumulation introduces latency and does not exploit fully the asynchronous nature and high temporal resolution of the sensor. To overcome the aforementioned problems, the nature of the sensor implies that each event should be dynamically augmented with its duration and velocity. Authors in [10] introduced the concept of an *event lifetime* and propose to estimate it by leveraging optical flow [11], thus allowing to identify a set of active events at any given time instance. An event is considered active as long as the change in brightness that triggered it is visible by its respective pixel. Using event lifetimes to identify the set of active events implicitly generates sharp gradient images at any point in time, thus effectively functioning as an edge detector.

Event-based vision sensors have also been proven successful in depth estimation via stereo matching. Some event-driven methods rely on standard stereo matching algorithms by generating artificial event frames. In [12], [13] authors used a fixed event accumulation interval to generate artificial frames, while [**?**] generated different frames based on event timestamps. A similar approach is to match events by comparing their local context by forming descriptors [14]. However, opposed to choosing a fixed time event accumulation interval, enhancing the event stream with event lifetimes yields more accurate disparity maps [15]. Therein the authors proposed to run stereo lifetime estimation separately for each sensor and subsequently estimate the disparity, which requires either using multiple threads or introduces latency caused by lifetime computation. Since the two sensors are mostly observing the same scene, they will also emit events triggered by the same brightness changes, thus offering the possibility to exploit the fact that the corresponding events should also have the same lifetimes.

In this paper we propose a novel method for event lifetime and disparity estimation in stereo event-cameras. Since events in a stereo setup are caused by the same change in brightness, we couple stereo event disparity and lifetime estimation in order to calculate them efficiently in a single shot. The resulting disparity estimation method is computationally more efficient and accurate in comparison to the method that decouples event lifetime and disparity estimation [15]. The proposed approach is validated on real-world data through multiple stereo event-camera experiments. The algorithm runs in a single thread, making it appropriate for energy-efficient applications in autonomous systems and mobile robotics.

The rest of the paper is organized as follows. In Section II we describe the event lifetime estimation method proposed in [10] based on optical flow and describe the local plane fitting algorithm. Our novel method for stereo event lifetime estimation together with disparity calculation is presented in Section III. Finally, we present the experimental results in Section IV and conclude with final remarks in Section V.

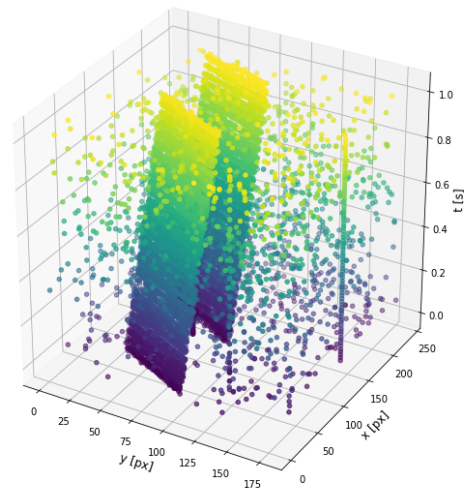


Fig. 2: Surface of active events for a moving line depicting events from both the left and right camera, where outliers are caused by sensor noise. Color gradient represents the timestamp for easier visibility.

II. Optical Flow Based Event Lifetime Estimation

Event lifetime estimation was proposed in [10] as a way to identify the set of active events at any given time instance, resulting in frames of accumulated events with clear structure and filtered noise that can be used as an input to standard computer vision algorithms for further processing. Herein we provide its description for completeness as it is an important part of the proposed stereo event lifetime and disparity estimation method.

A. Method Overview

The surface of active events (SAE) is a three-dimensional structure defined in the spatio-temporal domain where coordinates of each pixel of the sensor are mapped to the timestamp of the last event that occurred on that position (Figure 2). Formally, SAE can be described using a function $S : \mathbb{R}^2 \rightarrow \mathbb{R}, S(\mathbf{p}) = t$, where $\mathbf{p} = (x, y)^\top$ is the pixel position and t is the corresponding timestamp. The first order Taylor expansion gives the planar approximation of the SAE at a pixel location $\mathbf{p}$:

$$S(\mathbf{p} + \Delta\mathbf{p}) \approx S(\mathbf{p}) + (S_x(\mathbf{p}), S_y(\mathbf{p}))\, \Delta\mathbf{p}, \tag{1}$$

where S_x and S_y denote the first partial derivatives, $\frac{\partial S}{\partial x}$ and $\frac{\partial S}{\partial y}$, respectively. As proposed in [10], lifetime τ of the event at the point $(\mathbf{p}, t)$ is defined as the first order approximation of the maximum temporal increment of S for a displacement of one pixel. Therefore, the lifetime τ corresponds to the maximum amount of time before the brightness change at the current pixel will trigger a new event in a neighboring pixel. In other words, only one event caused by a single change in brightness should be considered active at some point in time; this can formally be expressed as:

$$\tau(\mathbf{p}) = \max\{\Delta t\}, \text{ subject to } \|\Delta\mathbf{p}\| = 1, \tag{2}$$

where $\Delta t = S(\mathbf{p} + \Delta\mathbf{p}) - S(\mathbf{p})$. Substitution further yields $\Delta t = (S_x(\mathbf{p}), S_y(\mathbf{p}))\,\Delta\mathbf{p} = \nabla S(\mathbf{p})\Delta\mathbf{p}$. Finally, lifetime τ can be expressed as:

$$\tau(\mathbf{p}) = \max\{\nabla S(\mathbf{p})\Delta\mathbf{p}\}, \text{ subject to } \|\Delta\mathbf{p}\| = 1. \quad (3)$$

Time t being a naturally increasing function, $S(\mathbf{p})$ is a monotonically increasing function of $\mathbf{p}$, thus having nonzero gradient at any point. This allows us to use the inverse function theorem around the location $\mathbf{p}$, as shown in [11]:

$$\begin{aligned} \frac{\partial S}{\partial x}(x, y_0) &= \frac{1}{v_x(x, y_0)}, \\ \frac{\partial S}{\partial y}(x_0, y) &= \frac{1}{v_y(x_0, y)}. \end{aligned} \quad (4)$$

The gradient can thus be expressed as:

$$\nabla S(\mathbf{p}) = \left(\frac{1}{v_x(\mathbf{p})}, \frac{1}{v_y(\mathbf{p})}\right)^\top, \quad (5)$$

showing its relation to the optical flow velocities, v_x and v_y. Assumption of constant velocities is equivalent to the local planar approximation of the SAE. The maximum temporal increment Δt occurs if the displacement $\Delta\mathbf{p}$ matches the unit vector in the direction of gradient $\Delta\mathbf{p} = \nabla S(\mathbf{p})/\|\nabla S(\mathbf{p})\|$. Therefore,

$$\tau(\mathbf{p}) = \nabla S(\mathbf{p})\frac{\nabla S(\mathbf{p})}{\|\nabla S(\mathbf{p})\|} = \sqrt{v_x^{-2} + v_y^{-2}}. \quad (6)$$

The normal of the tangent plane to the SAE at the point $\mathbf{p}$ is given by

$$\begin{aligned} \mathbf{n}(\mathbf{p}) &\propto (1, 0, S_x(\mathbf{p}))^\top \times (0, 1, S_y(\mathbf{p}))^\top \\ &= (-S_x(\mathbf{p}), -S_y(\mathbf{p}), 1)^\top. \end{aligned} \quad (7)$$

Furthermore, combination of (5) and (7) yields $\mathbf{n}(\mathbf{p}) \propto (-v_x^{-1}, -v_y^{-1}, 1)^\top$, and naming the normal coordinates as $\mathbf{n}(\mathbf{p}) = (n_1, n_2, n_3)^\top$ implies $-v_x^{-1} = \frac{n_1}{n_3}$ and $-v_y^{-1} = \frac{n_2}{n_3}$. Components of the SAE tangent plane normal $(n_1, n_2, n_3)^\top$ are extracted by fitting a plane to the data using an optimization technique (see Section II-B) that finally enables computation of the lifetime $\tau(\mathbf{p})$ as:

$$\tau(\mathbf{p}) = \frac{1}{n_3}\sqrt{n_1^2 + n_2^2}. \quad (8)$$

B. Local plane-fitting Algorithm

A plane-fitting algorithm is needed to calculate the parameters of the SAE tangent plane to further estimate the lifetime of the given event at the location $\mathbf{p}$. In [11], it was proposed to use an $N \times N \times 2\Delta t$ window centered around the current event to estimate the local plane, including the events from the future. To avoid the introduced Δt latency, we use only the past events in the $N \times N$ spatial window based on the local smoothness assumption, as suggested by [10]. Even though therein they completely eliminate the Δt parameter and the need to tune it, we use $\Delta t = 100\,\text{ms}$ as a time frame in which the past events in the spatial window are considered to ensure that the events that are too old and unrelated to recent events do not affect the estimated tangent plane negatively.

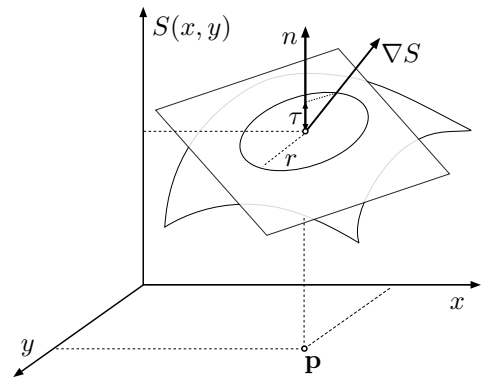


Fig. 3: Principle of optical flow computation using SAE. Planar approximation of SAE is used to compute lifetimes of the events. Radius r equals 1.

Fitting the plane using the RANSAC algorithm suppresses noise events introduced by the sensor and ensures robustness. First, a plane is fitted to a set of hypothetical inliers, which include the current event and two random past events in the spatio-temporal window. All other past events are then tested against the fitted plane. They are considered as inliers if their distance to the plane is below the threshold μ. If less than m inliers are found, the process is repeated from the beginning with a new hypothetical set of inliers. The event is declared as noise if the condition for the minimum number of inliers is not met in the predefined maximum number of iterations. Otherwise, the plane is reestimated by fitting to all the inliers using the least squares minimization:

$$\mathbf{n}_{\text{LS}} = \arg\min_{\mathbf{n}} \|\mathbf{A}\mathbf{n} - \mathbf{b}\|^2, \quad (9)$$

where $\mathbf{A}$ is the $n \times 3$ matrix of the n SAE points (x_i, y_i, t_i) identified as inliers, and $\mathbf{b} = (1 \cdots 1)^\top$.

Assumption of locally constant velocity allows for prediction of the future events timestamp, as suggested in [10], wherein authors proposed to use the predicted timestamp to compute the regularization factor depending on the absolute difference of the predicted timestamp $\hat{t}_i$ and the actual timestamp t_i of the incoming events:

$$\Delta t_{\text{err}} = |t_i - \hat{t}_i|. \quad (10)$$

That method performs plane fitting based on the RANSAC algorithm for each event, even if the predicted plane normal is considered very reliable. The more the predicted and actual timestamp are close, the more the plane normal obtained from the previous events is considered reliable. Our method, however, avoids least squares optimization for the events whose timestamps closely match the predicted ones. In that case, we assume that the normal of the current event tangent plane is equal to the predicted normal, saving computational time while retaining accuracy.

III. Stereo Event Lifetime and Disparity Estimation

In this section we propose a method for estimating the lifetimes of events from two dynamic vision sensors in stereo settings. Naturally, streams of events emitted by the two sensors are very similar since they mostly observe the same scene. For the same brightness change, both sensors are triggered and emit an event. Therefore, it makes sense to calculate the lifetime of those events only once per brightness change and associate it with the other pertaining event as well.

As Algorithm 1 shows, for the current event, we attempt to find its counterpart that was caused by the same brightness gradient in the SAE of the other sensor, which is equivalent to finding its disparity. We calculate the disparity for each event *asynchronously*, at the time of its arrival. Even though our approach is event-driven, the method is similar to the standard stereo matching methods in a sense that it operates on frames of accumulated events. We use an event feature descriptor introduced by [15] that supports event-driven stereo matching based on distance transform [16] which describes the context of a pixel in binary images by calculating the distance from the pixel to the nearest active pixel. To apply it to the problem of an event descriptor, coordinates of active events at the time of the current event's arrival are considered as active pixels, and others as inactive. It is important to note that we make an assumption that the current event is certainly active at its starting point. Also, even if some future events (relative to the current event) from the other sensor have already been processed, we cannot consider them as active since their starting timestamp is later than the timestamp of the current event. As a starting step of descriptor computation, we calculate the vector pointing to the nearest active event for every pixel in a $N \times N$ window centered at the current event's position $\mathbf{p}$, separately for events of positive and negative polarity. Then, we classify the obtained vectors by orientation. We assign each vector to one of the 12 bins of the histogram, each bin covering a $30°$ interval. The final feature descriptor $\mathbf{D}$ of an event is then defined as a vector of 24 values; 12 bins of the histogram for the positive, and likewise for the negative events. Although [15] proposed a descriptor invariant to scale and rotation, we do not implement those features since they are not needed for stereo matching on rectified images.

Disparity map is computed in three standard stereo matching steps: *i)* matching cost calculation, *ii)* cost aggregation, and *iii)* disparity estimation. Matching cost of an event in location $\mathbf{p}$ for disparity d is calculated by summing absolute differences between the corresponding event descriptors $\mathbf{D}^L$ and $\mathbf{D}^R$, from left and right cameras respectively, within a rectangular window W_p centered at $\mathbf{p}$:

$$C(\mathbf{p}, d) = \sum_{(x_i, y_i) \in W_p} \left| \mathbf{D}^L(x_i, y_i) - \mathbf{D}^R(x_i - d, y_i) \right|. \quad (11)$$

Cost aggregation is performed as suggested in [15], in a fixed region A around the location $\mathbf{p}$:

$$C_A^0(\mathbf{p}, d) = C(\mathbf{p}, d),\ C_A^i(\mathbf{p}, d) = \sum_{\mathbf{q} \in A_p} C_A^{i-1}(\mathbf{q}, d). \quad (12)$$

Finally, disparity estimation is done through the winner–takes–all approach, by finding the disparity d with the minimal aggregated cost:

$$D(\mathbf{p}) = \arg\min_{\mathbf{d}} C_A(\mathbf{p}, d). \quad (13)$$

In case of successful disparity estimation, we determine the current event's lifetime as a median of lifetimes of the events in the matched spatial window M:

$$\tau(\mathbf{p}) = \mathrm{median}\{\tau(\mathbf{q}), \mathbf{q} \in M\}. \quad (14)$$

Other computationally light operations for calculating the lifetime using the matched window can be considered as well, e.g., averaging, maximum, minimum, or similar. We chose median because it is least prone to outliers. In case of failed disparity estimation, we estimate the lifetime for the event using the aforementioned method.

There are two main advantages of this method opposed to calculating the lifetimes of the events from the two sensors separately. First, this method is less computationally complex compared to performing the RANSAC algorithm for plane fitting for each event, as will demonstrated in the Section IV. Furthermore, due to noise and low resolution of the sensor, it is not possible to determine the lifetime of each event using the plane fitting method, so they are falsely declared as noise. In that case, however, it might be possible to determine the lifetime of the corresponding event in the other sensor's SAE, allowing to retain the event that would otherwise be falsely proclaimed as noise, thus generating a more finely detailed image with sharper gradients. The method is suitable for execution in a single thread, making it convenient to implement in systems with limited computational resources. It is also important to note that our method offers high level of abstraction, i.e., it is possible to use any suitable lifetime estimation and disparity estimation method other than the proposed ones depending on the application.

Algorithm 1 Stereo event lifetime ($sensor_A$, $sensor_B$)

```
read set of events E_A from sensor_A
for e_A(x_i, y_i, t_i) in E_A do
  if event e_B(_, _, t_i) present in E_B then
    d = computeDisparity(x_i, y_i, t_i)
    if d > 0 then
      lifetime = medianLifetime(e_B)
    else
      lifetime = computeLifetime(e_A)
    end if
  else
    lifetime = computeLifetime(e_A)
  end if
  setLifetime(e_A, lifetime)
end for
```

IV. Experimental Results

We evaluate the proposed method on multiple real-world stereo experiments. Our algorithm's output consists of the disparity map and the image of accumulated events for each of the sensor views. Since there is no metric defined to evaluate the accuracy of lifetime estimation, we compare the outputs qualitatively. We compare the proposed method in terms of both execution time and accuracy with the method proposed in [15] that first executes lifetime estimation separately for both sensors and then follows up with disparity estimation (referred to as the *decoupled method*). We also compare our output with the method that uses a fixed event accumulation interval, i.e., does not estimate event lifetime. As future work, the proposed method can be evaluated on the complex real-world stereo dynamic vision sensor dataset [**?**].

A. Experimental Setup

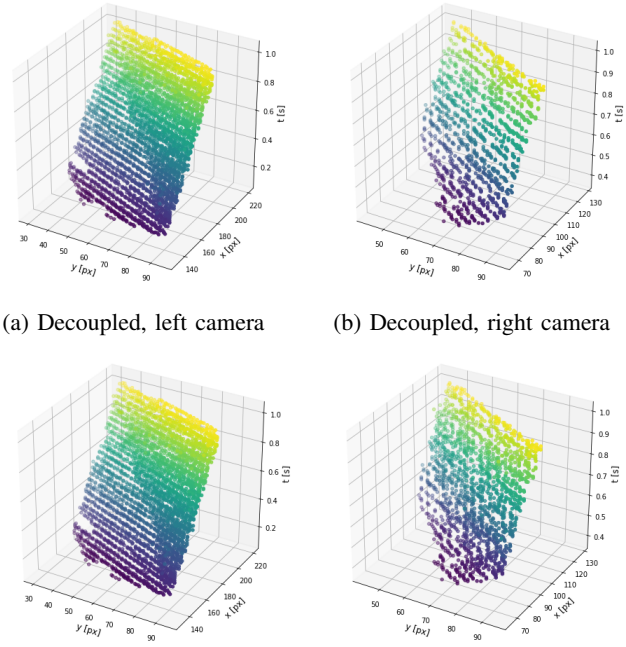
Experiments were conducted with two DAVIS240 sensors mounted on a stereo rig with a baseline of 10 cm as shown in Fig. 1d. Resolution of the sensors is 240×180 pixels. Timestamps of the events were synchronized on the hardware level. Event data was undistorted and stereo rectified using the Kalibr toolbox [18]. We used the DAVIS240 standard camera frames for calibration, but our proposed method operates on events only. Experiments were run on a personal computer with Intel Core i7-7700HQ CPU at 2.80GHz on a single core.

B. Implementation Details

The implementation of the proposed method is done in C++ in ROS environment [17]. Stream of events from DAVIS240 is fetched in grouped sequences through a ROS topic. A single thread is subscribed to messages from both left and right sensor. Random scheduling of the incoming ROS messages does not guarantee that the events from one sensor will always be available before the corresponding events from the other sensor, making it impossible to set one sensor as primary, and other as secondary. However, for the current set of events, it is possible to determine whether the corresponding events from the other sensor have already been processed and then decide whether to perform lifetime or disparity estimation first.

C. Accuracy Improvement

DVS yield sparse event data in case of a motion occurring within the area of very similar brightness intensity, which the sensor is unable to discern, resulting in poor results of event lifetime and disparity estimation. Figure 4 shows events obtained by recording a moving line from a computer screen. Due to sparse event data caused by screen flickering, the decoupled lifetime estimation method was not able to compute the lifetime of some events from the right sensor, thus proclaiming many events as noise. However, as the two sensors are naturally not completely identical in terms of sensitivity to brightness changes, the method was more successful in computing the lifetimes of events from the left sensor. Since the proposed stereo event lifetime computation method also uses events from the left sensor, it was able to compute lifetimes for more events from the right sensor compared to the decoupled lifetime computation, thus proving better accuracy in terms of noise detection compared to the decoupled method. Experiment shown in Figure 6 confirms that our method also matches the decoupled method in terms of accuracy in case of dense event data. Figures 1 and 5 show the experiments performed on real-world data comparing the proposed method to using a fixed event accumulation time interval of 10 ms, which yields very sparse gradient image lacking structure.



(a) Decoupled, left camera (b) Decoupled, right camera

(c) Proposed, left camera (d) Proposed, right camera

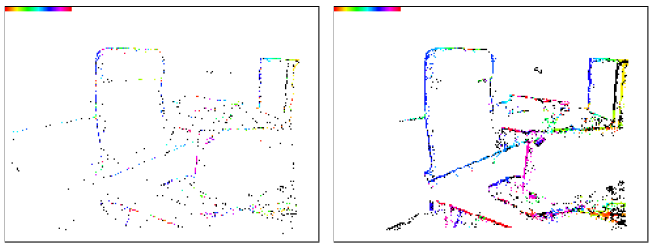
Fig. 4: SAEs of the events with successfully estimated lifetime obtained by observing a moving line (all events shown in Figure 2). Decoupled method falsely detects many events of the right camera as noise due to sparsity. The proposed method is able to provide lifetime for more events (1642) than the decoupled method (1178), thus providing better accuracy for the right camera, while retaining accuracy for the left camera.

D. Execution Time Improvement

Since the plane-fitting algorithm used to estimate the event lifetime is computationally complex, our method aims to reduce the number of times it needs to be executed as much as possible. As a first way of reducing execution time, we propose to use the previously calculated and predicted lifetime, if the prediction was reliable, without fitting a new plane. Furthermore, if the predicted lifetime is not available, our method executes plane-fitting only once for the two corresponding events caused by one brightness change, effectively cutting the execution time in half. The amount of improvement depends on the number of events whose disparity is successfully estimated, in which case we use the efficient method of calculating the current event lifetime by finding the median of lifetimes of the events in the matched window. Experiments show that the execution time is reduced approximately by 50% (see Table I).

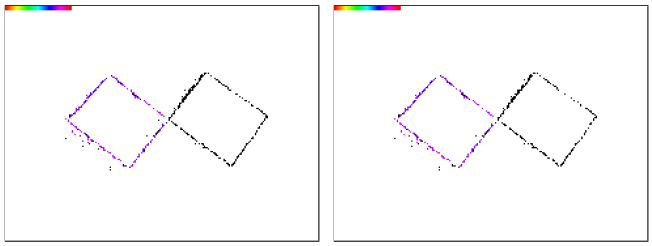
TABLE I: Number of events with estimated lifetime and pertaining estimation execution times. The proposed method runs nearly twice as fast compared to the decoupled method. The time needed to calculate the disparity is not included.

	Decoupled		Proposed	
Scene	Events	Time [ms]	Events	Time [ms]
Figure 1	2692	32	2759	16
Figure 5	2536	30	2498	15
Figure 6	426	11	436	6



(a) Fixed accumulation interval (b) Proposed method

Fig. 5: Comparison with respect to a fixed accumulation time interval of 10 ms. Disparity estimation is not reliable for nearly horizontal lines and areas prone to noise due to lack of structure, but our method still yields sharp gradient images.



(a) Decoupled method (b) Proposed method

Fig. 6: Image shows events from both sensors. Although both decoupled and proposed method yield sharp gradients and correct disparity in this example, our method performs nearly twice as fast compared to the decoupled approach.

V. Conclusion

In this paper, we have presented a novel method for stereo lifetime and disparity estimation for event-based cameras. Enhancing the events with their lifetime prior to disparity estimation makes the disparity maps representative of the scene structure invariant to the amount of motion of the sensor or the scene. Since one brightness change triggers an event in both of the sensors, we have proposed to calculate the lifetime for only one of those events, and associate it to the other one via stereo matching, while simultaneously determining its disparity. The proposed method is twice as fast as the method calculating the event lifetimes for each sensor independently, while it also provides disparity maps with more accurate structure, as we have demonstrated through several real-world experiments.

References

[1] T. Delbrück, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 2426–2429.

[2] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[3] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[4] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.

[5] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," *arXiv preprint arXiv:1904.08298*, 2019.

[6] L. Wang, Y.-S. Ho, K.-J. Yoon *et al.*, "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," *arXiv preprint arXiv:1811.08230*, 2018.

[7] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Régnier, "Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1081–1089, 2012.

[8] P. Bardow, A. J. Davison, and S. Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 884–892.

[9] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[10] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *2015 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4874–4881.

[11] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 2, pp. 407–417, 2014.

[12] S. Schraml, A. N. Belbachir, N. Milosevic, and P. Schön, "Dynamic stereo vision system for real-time tracking," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 1409–1412.

[13] J. Kogler, C. Sulzbachner, M. Humenberger, and F. Eibensteiner, "Address-event based stereo vision with bio-inspired silicon retina imagers," in *Advances in theory and applications of stereo vision*. IntechOpen, 2011.

[14] D. Zou, P. Guo, Q. Wang, X. Wang, G. Shao, F. Shi, J. Li, and P.-K. Park, "Context-aware event-driven stereo matching," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1076–1080.

[15] D. Zou, F. Shi, W. Liu, J. Li, Q. Wang, P. Park, C. Shi, Y. Roh, and H. Ryu, "Robust dense depth map estimation from sparse dvs stereos," in *British Machine Vis. Conf.(BMVC)*, vol. 3, 2017.

[16] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[18] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 473–480.